Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method for balancing the load of a parallel processing system having a plurality of parallel processing elements arranged in a loop, wherein each processing element (PE_r) has a local number of tasks associated therewith, wherein r represents the number for a selected processing element, and wherein each of said processing elements is operable to communicate with a clockwise adjacent processing element and with an anti-clockwise adjacent processing element, the method comprising:

determining within each processing element a total number of tasks present within said loop;

calculating a local mean number of tasks within each of said plurality of processing elements;

calculating a local deviation from said local mean number within each of said plurality of processing elements;

determining a sum deviation from said local deviations within each of said processing elements for one-half said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

determining a sum deviation from said local deviations within each of said processing elements for one-half said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

determining a clockwise transfer parameter and an anti-clockwise transfer parameter from said sum deviations within each of said processing elements; and

redistributing tasks among said plurality of processing elements in response to said clockwise transfer parameters and said anti-clockwise parameters within each of said plurality of processing elements.

2. (original) The method of claim 1 wherein said determining within each of said processing elements a total number of tasks present within said loop, comprises:

transmitting said local number of tasks associated with each of said processing elements to each other of said plurality of processing elements within said loop;

receiving within each of said processing elements said number of local tasks associated with said each other of said plurality of processing elements; and

summing said number of local tasks associated with each of said processing elements with said number of local tasks associated with each other of said plurality of processing elements.

3. (original) The method of claim 1 wherein said determining said total number of tasks present within said loop includes solving the equation $V = \sum_{i=0}^{i=N-1} v_i$, where N represents the number of processing elements in said loop, and v_i represents said local number of tasks associated with an i^{th} processing element in said loop.

4. (currently amended) The method of claim 1 wherein said calculating a local mean number of tasks within each of said plurality of processing elements includes solving the equation $M_r = Trunc((V + E_r)/N)$, where M_r is said local mean for PE_r, N is the total number of processing elements in said loop, and E_r is a number in the range of 0 to $(N-1)$, V is the total number of tasks, and wherein each processing element has a different E_r value.

5. (currently amended) The method of claim [[3]] 4 wherein ~~E_r controls~~ said *Trunc* function is responsive to the value of E_r such that said total number of tasks for said loop is equal to the sum of the local mean number of tasks for each of said plurality of processing elements in said loop (i.e., $V = \sum_{i=0}^{i=N-1} M_i$).

6. (currently amended) The method of claim [[3]] 4 wherein said local mean $M_r = Trunc((V + E_r) / N)$ for each local PE_r within said loop is equal to either ~~one of~~ X ~~and~~ or $(X+1)$, and E_r is a number in the range of 0 to $(N-1)$.

7. (original) The method of claim 1 wherein said calculating a local deviation within each of said plurality of processing elements, comprises finding the difference between said local number of tasks and said local mean number for each of said plurality of processing elements.

8. (original) The method of claim 1 wherein said determining a sum deviation within each of said processing elements for one-half of said loop in an anti-clockwise direction comprises:

transmitting said local deviation associated with each of said processing elements half way around said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local deviation associated with each other of said processing elements half way around said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing said local deviations associated with each other of said processing elements half way around said loop in a clockwise direction.

9. (original) The method of claim 1 wherein said determining a sum deviation within each of said processing elements in one-half of said loop in a clockwise direction comprises:

transmitting said local deviation associated with each of said processing elements half way around said loop in an clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local deviation associated with each other of said processing elements half way around said loop in a anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing said local deviations associated with each other of said processing elements half way around said loop in an anti-clockwise direction.

10. (currently amended) The method of claim 1 wherein said determining a clockwise transfer parameter and an anti-clockwise transfer parameter within each of said processing elements comprises:

setting the clockwise transfer parameter equal to $(2S + A - C) \div 4$; and

setting the anti-clockwise transfer parameter equal to $(2S + C - A) \div 4$, where S represents ~~the~~ said local deviation of a selected processing element; C represents said sum deviation in a clockwise half of loop relative to said selected processing element, and A represents said sum deviation in an anti-clockwise half of loop relative to said selected processing element.

11. (currently amended) The method of claim 1 wherein said determining a clockwise transfer parameter T_c and an anti-clockwise transfer parameter T_a within each of said processing elements comprises at least one of:

setting the clockwise transfer parameter equal to $Trunc[(2S + \Delta) \div 4]$ and setting the anti-clockwise transfer parameter equal to $S - T_c$ and

setting the anti-clockwise transfer parameter equal to $Trunc[(2S - \Delta) \div 4]$ and setting the clockwise transfer parameter equal to $S - T_a$;

where ~~Δ = Mag, if Δ > Mag, where Δ = – Mag, if Δ < -Mag, where~~ Mag = abs(2S), ~~and where~~ S represents the local deviation of a selected processing element, Δ represents the number of tasks passing through the current processing element, whereby if Δ > Mag then set Δ equal to Mag and if Δ < -Mag, then set Δ equal to –Mag.

12. (currently amended) A method for reassigning tasks among an odd numbered plurality of processing elements within a parallel processing system, said processing elements being connected in a loop and each having a local number of tasks associated therewith, the method comprising:

determining the total number of tasks on said loop;

computing a local mean value for a selected processing element;

computing a local deviation for said selected processing element, said local deviation representative of the difference between said local number of tasks for said selected processing element and said local mean value for said selected processing element;

inserting a phantom processing element within said loop having a local deviation of zero when the loop is comprised of an odd number of processing elements;

summing said local deviations ~~deviation~~ of said processing elements located within one-half of the loop in an anti-clockwise direction relative to said selected processing element;

summing said local deviations ~~deviation~~ of said processing elements located within one-half of the loop in a clockwise direction relative to said selected processing element;

computing a number of tasks to transfer in a clockwise direction for said selected processing element in response to said summing of said local deviations;

computing a number of tasks to transfer in an anti-clockwise direction for said selected processing element in response to said summing of said local deviations; and

reassigning tasks relative to the said number of tasks to transfer in a clockwise direction and said number of task to transfer in an anti-clockwise direction.

13. (original) The method of claim 12 wherein said determining the total number of tasks on said loop, comprises:

transmitting said local number of tasks associated with each of said processing elements to each other of said plurality of processing elements within said loop;

receiving within each of said processing elements said number of local tasks associated with said each other of said plurality of processing elements; and

summing said number of local tasks associated with each of said processing elements with said number of local tasks associated with each other of said plurality of processing elements.

14. (currently amended) The method of claim 12 wherein computing a local mean value for a selected processing element includes solving the equation $M_r = Trunc((V + E_r)/N)$, where

M_r is said local mean for a processing element PE_r, N is the total number of processing elements in said loop, V is the total number of tasks, and E_r is a number in the range of 0 to $(N-1)$.

15. (currently amended) The method of claim 14 wherein ~~E_r controls~~ said *Trunc* function is responsive to the value of E_r such that said total number of tasks for said loop is equal to the sum of the local mean number of tasks for each of said plurality of processing elements in said loop and wherein each processing element has a different E_r value assigned.

16. (currently amended) The method of claim 12 wherein said inserting a phantom processing element within said loop further comprises:

locating said phantom processing element in a position within said loop that is diametrically opposed to said processing element~~; and~~

~~assigning a zero deviation value to said phantom processing element~~.

17. (original) The method of claim 12 wherein said computing a local mean value for a selected processing element, said computing a local deviation for said selected processing element, said inserting a phantom processing element within said loop, said summing said deviation of said processing elements located within one-half of the loop in an anti-clockwise direction, summing said deviation of said processing elements located within one-half of the loop in a clockwise direction, computing a number of tasks to transfer in a clockwise direction for said selected processing element, computing a number of tasks to transfer in an anti-clockwise direction for said selected processing element, and reassigning tasks relative to the said number of task to transfer in a clockwise direction and said number of tasks to transfer in an anti-clockwise direction are completed simultaneously for each of said plurality of processing elements within said loop.

18. (original) The method of claim 12 wherein said summing said deviation of said processing elements located within one-half of the loop in an anti-clockwise direction relative to said selected processing element comprises:

transmitting said local deviation associated with each of said processing elements half way around said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local deviation associated with each other of said processing elements half way around said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing said local deviations associated with each other of said processing elements half way around said loop in a clockwise direction.

19. (original) The method of claim 12 wherein summing said deviation of said processing elements located within one-half of the loop in a clockwise direction relative to said selected processing element comprises:

transmitting said local deviation associated with each of said processing elements half way around said loop in an clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local deviation associated with each other of said processing elements half way around said loop in a anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing said local deviations associated with each other of said processing elements half way around said loop in an anti-clockwise direction.

20. (currently amended) A computer readable memory device carrying a set of instructions which, when executed, perform a method comprising:

determining within each processing element a total number of tasks present within said loop;

calculating a local mean number of tasks within each of said plurality of processing elements;

calculating a local deviation from said local mean number within each of said plurality of processing elements;

determining a sum deviation from said local deviations within each of said processing elements for one-half said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

determining a sum deviation from said local deviations within each of said processing elements for one-half said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

determining a clockwise transfer parameter and an anti-clockwise transfer parameter from said sum deviations within each of said processing elements; and

redistributing tasks among said plurality of processing elements in response to said clockwise transfer parameters and said anti-clockwise parameters within each of said plurality of processing elements.